# Figure 1
AUS990853

108

Client

110

Client

Network
102

104

Server

112

Client

Storage

100

106

Processor
202

Processor
204

206

System Bus

208 — Memory
Controller/
Cache

I/O Bridge — 210

212

Local Memory
209

216

PCI Bus Bridge
214

PCI Bus

218

220

Modem

Network
Adapter

I/O Bus

Graphics
Adapter
230

226

PCI Bus Bridge
222

PCI Bus

228

Hard Disk
232

PCI Bus Bridge
224

PCI Bus

200

# Figure 2A

AUS990853

Processor
252

Host/PCI
Cache/Bridge
258

Main Memory
254

Audio Adapter
266

Bus 256

SCSI Host
Bus
Adapter
262

LAN
Adapter
260

Expansion
Bus Interface
264

Graphics
Adapter
268

Audio/
Video
Adapter
269

Disk 276

Tape 278

CD-
ROM
280

DVD 282

Keyboard and
Mouse
Adapter
270

Modem
272

Memory
274

250
Client

Figure 2B

AUS990853

# Figure 3A

Java Virtual Machine
(JVM)
304

Platform Specific
Operating System
302

Java Application/Applet
306
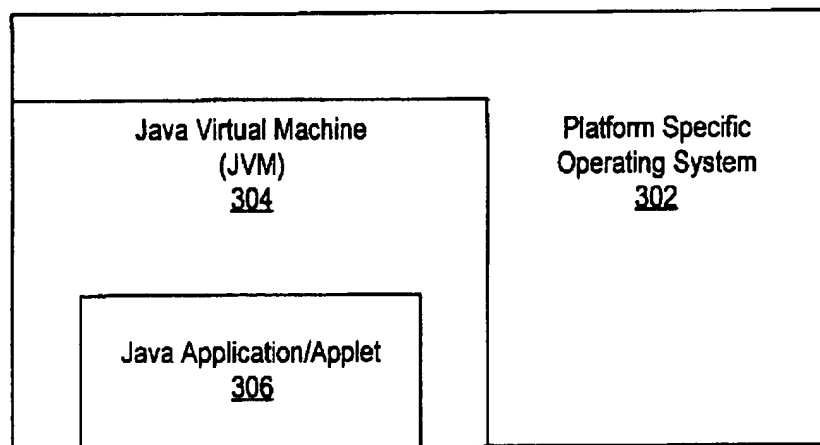
300

# Figure 3B

AUS990853US1

350

Class
Files

Class Loader
Subsystem
352

Runtime Data Areas
354

Method
Area
370

Heap
372

Java
Stacks
366

PC
Registers
368

Native
Method
Stacks
364

Memory
Management
374

Native Method
Interface
358

Native
Method
Libraries

Execution Engine
356

Compiler
360

Interpreter
362

# Figure 4

AUS990853

```
┌──────────────┐                          ┌──────────────┐
│    Trace     │ ──────────────────────►  │   Process.   │
│  Program.    │                          │     402      │
│     400      │ ◄──────────────────────  │              │
└──────────────┘                          └──────────────┘
        │
        │
        ▼
┌──────────────┐
│   Buffer.    │
│     404      │
└──────────────┘
```

```
Begin.
```

```
Initialization
Phase
500
```

```
Profiling Phase
502
```

```
Post-
Processing
Phase
504
```

```
End.
```

# Figure 5

# Figure 6

AUS990853

```
        ┌───────────┐
        │  Begin.   │
        └─────┬─────┘
              │
              ▼
     ┌─────────────────┐         ┌─────────────────┐
     │ Allocate Buffer │         │  Start Tracing  │
     │       600       │         │       604       │
     └────────┬────────┘         └────────┬────────┘
              │                           │
              ▼                           ▼
     ┌─────────────────┐         ┌─────────────────┐
     │  Turn On Trace  │         │  Receive Trace  │
     │      Hooks      │         │      Data       │
     │       602       │         │       606       │
     └────────┬────────┘         └────────┬────────┘
              │                           │
              │                           ▼
              └──────────────┐   ┌─────────────────┐
                             │   │   Store Trace   │
                             │   │  Data in Buffer │
                             │   │       608       │
                             │   └────────┬────────┘
                                          │
                                          ▼
                                    ◇─────────────◇
                                   ╱   Tracing     ╲      No
                                  ◇   Finished?     ◇────────┐
                                   ╲     610       ╱         │
                                    ◇─────────────◇          │
                                          │                  
                                          │ Yes              
                                          ▼                  
                                 ┌─────────────────┐
                                 │  Send Buffer    │
                                 │ Contents to File│
                                 │       612       │
                                 └────────┬────────┘
                                          │
                                          ▼
                                 ┌─────────────────┐
                                 │    Generate     │
                                 │     Report      │
                                 │       614       │
                                 └────────┬────────┘
                                          │
                                          ▼
                                    ┌───────────┐
                                    │   End.    │
                                    └───────────┘
```

# Figure 7

Interrupt Hook

```
        ┌─────────────┐
        │   Begin.    │
        └─────────────┘
               │
               ▼
      ┌──────────────────┐
      │ Obtain Program   │
      │    Counter       │
      │      700         │
      └──────────────────┘
               │
               ▼
      ┌──────────────────┐
      │ Obtain Interrupted│
      │    Thread ID     │
      │      701         │
      └──────────────────┘
               │
               ▼
            ◇─────────◇
          ◇   Interpreter  ◇        No
         ◇     Code?        ◇──────────────┐
          ◇    702        ◇               │
            ◇─────────◇                   │
               │ Yes                      │
               ▼                          │
      ┌──────────────────┐                │
      │ Identify Method  │                │
      │  Block Being     │                │
      │  Interpreted     │                │
      │      704         │                │
      └──────────────────┘                │
               │                          │
               ▼                          │
      ┌──────────────────┐                │
      │   Send Trace     │◄───────────────┘
      │  Information     │
      │      706         │
      └──────────────────┘
               │
               ▼
        ┌─────────────┐
        │    End.     │
        └─────────────┘
```

# Figure 8

# Figure 9

AUS990853

900

| PID | TID | addr1 | addr2 | ... | addrN |
|-----|-----|-------|-------|-----|-------|

902  904  906  908  910

| Timestamp | Event | Call Stack After Event |
|---|---|---|
| 0 | enter C | |
| 1 | enter A | C |
| 2 | enter B | CA |
| 3 | exit from B | CAB |
| 4 | enter B | CA |
| 5 | enter B | CAB |
| 6 | exit from B | CABB |
| 7 | exit from B | CAB |
| 8 | exit from A | CA |
| 9 | enter B | C |
| 10 | enter A | CB |
| 11 | enter B | CBA |
| 12 | enter A | CBAB |
| 13 | exit from A | CBABA |
| 14 | exit from B | CBA |
| 15 | enter X | CBAX |
| 16 | exit from X | CBA |
| 17 | exit from A | CB |
| 18 | exit from B | C |
| 19 | exit from C | |

## Figure 10A

AUS990853

| Sample | Call Stack @ Sample |
|---|---|
| 1 | C |
| 2 | CAB |
| 3 | CAB |
| 4 | CAB |
| 5 | C |
| 6 | CBA |
| 7 | CBABA |
| 8 | CBA |
| 9 | CBA |
| 10 | C |

## Figure 10B

AUS990853
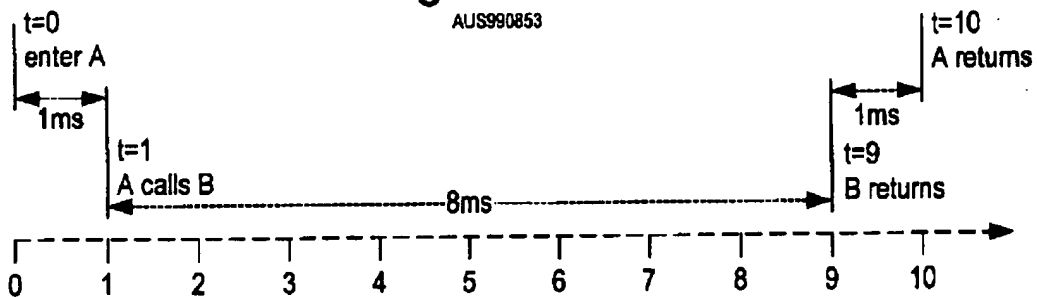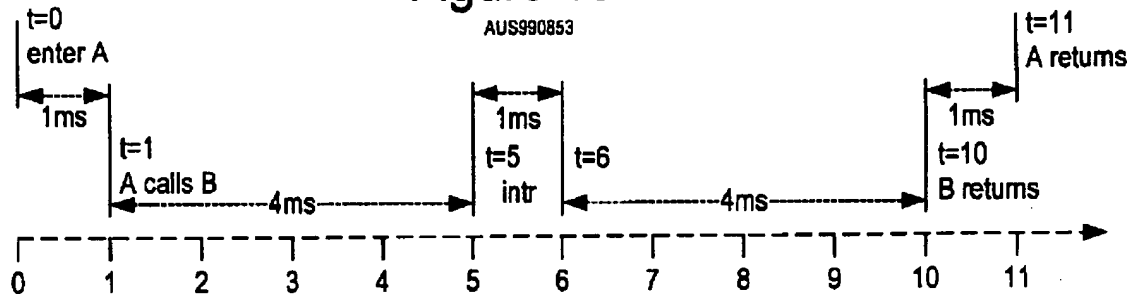
# Figure 10C

t=0
enter A

1ms

t=1
A calls B

8ms

t=10
A returns

1ms

t=9
B returns

```
|----|----|----|----|----|----|----|----|----|----|---->
0    1    2    3    4    5    6    7    8    9    10
```

# Figure 10D

t=0
enter A

1ms

t=1
A calls B

4ms

1ms
t=5
intr

t=6

4ms

t=11
A returns

1ms

t=10
B returns

```
|----|----|----|----|----|----|----|----|----|----|----|--->
0    1    2    3    4    5    6    7    8    9    10   11
```

# Figure 11A

1100

```
                                    ┌──────────────┐
                                    │ C: Addr      │
                                    │ BASE         │
                                    │ CUM          │
                                    │ Parent       │
                                    │ Children     │
                                    │ 1102         │
                                    └──────────────┘
                                   /              \
                    ┌──────────────┐          ┌──────────────┐
                    │ A: Addr      │          │ B: Addr      │
                    │ BASE         │          │ BASE         │
                    │ CUM          │          │ CUM          │
                    │ Parent       │          │ Parent       │
                    │ Children     │          │ Children     │
                    │ 1106         │          │ 1104         │
                    └──────────────┘          └──────────────┘
                          /                          \
            ┌──────────────┐                    ┌──────────────┐
            │ B: Addr      │                    │ A: Addr      │
            │ BASE         │                    │ BASE         │
            │ CUM          │                    │ CUM          │
            │ Parent       │                    │ Parent       │
            │ Children     │                    │ Children     │
            │ 1116         │                    │ 1108         │
            └──────────────┘                    └──────────────┘
                                                      \
                                              ┌──────────────┐
                                              │ B: Addr      │
                                              │ BASE         │
                                              │ CUM          │
                                              │ Parent       │
                                              │ Children     │
                                              │ 1110         │
                                              └──────────────┘
                                                    /
                                          ┌──────────────┐
                                          │ A: Addr      │
                                          │ BASE         │
                                          │ CUM          │
                                          │ Parent       │
                                          │ Children     │
                                          │ 1114         │
                                          └──────────────┘
```
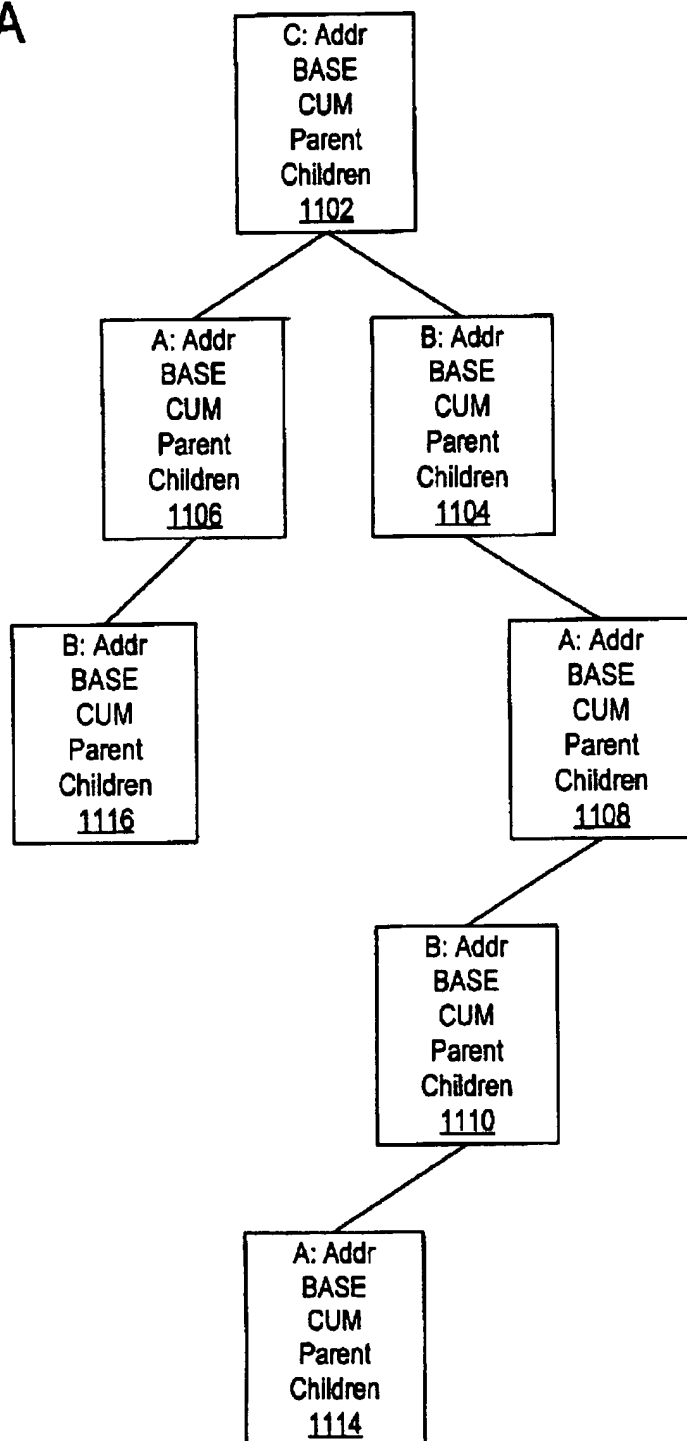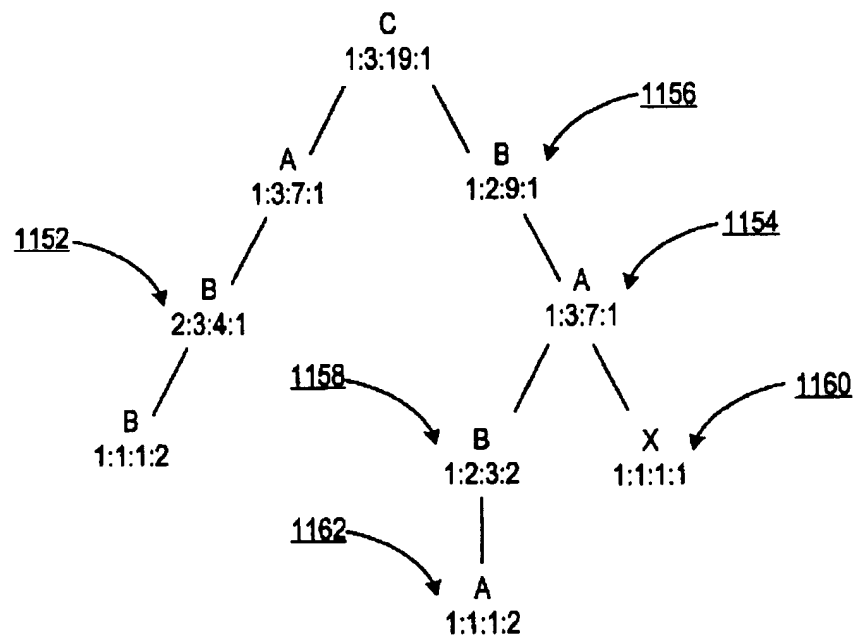
**Figure 11B**
AUS990853

# Figure 12

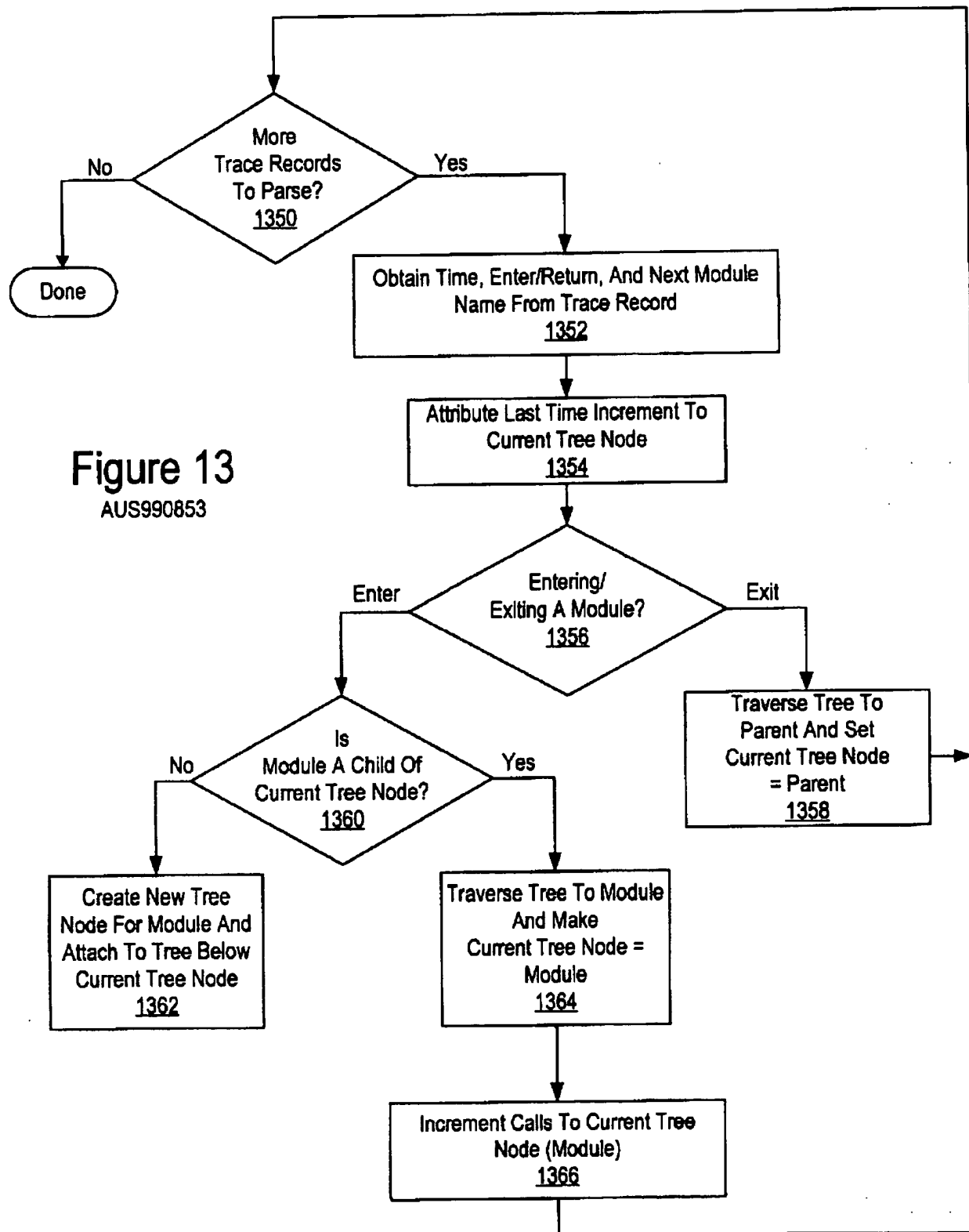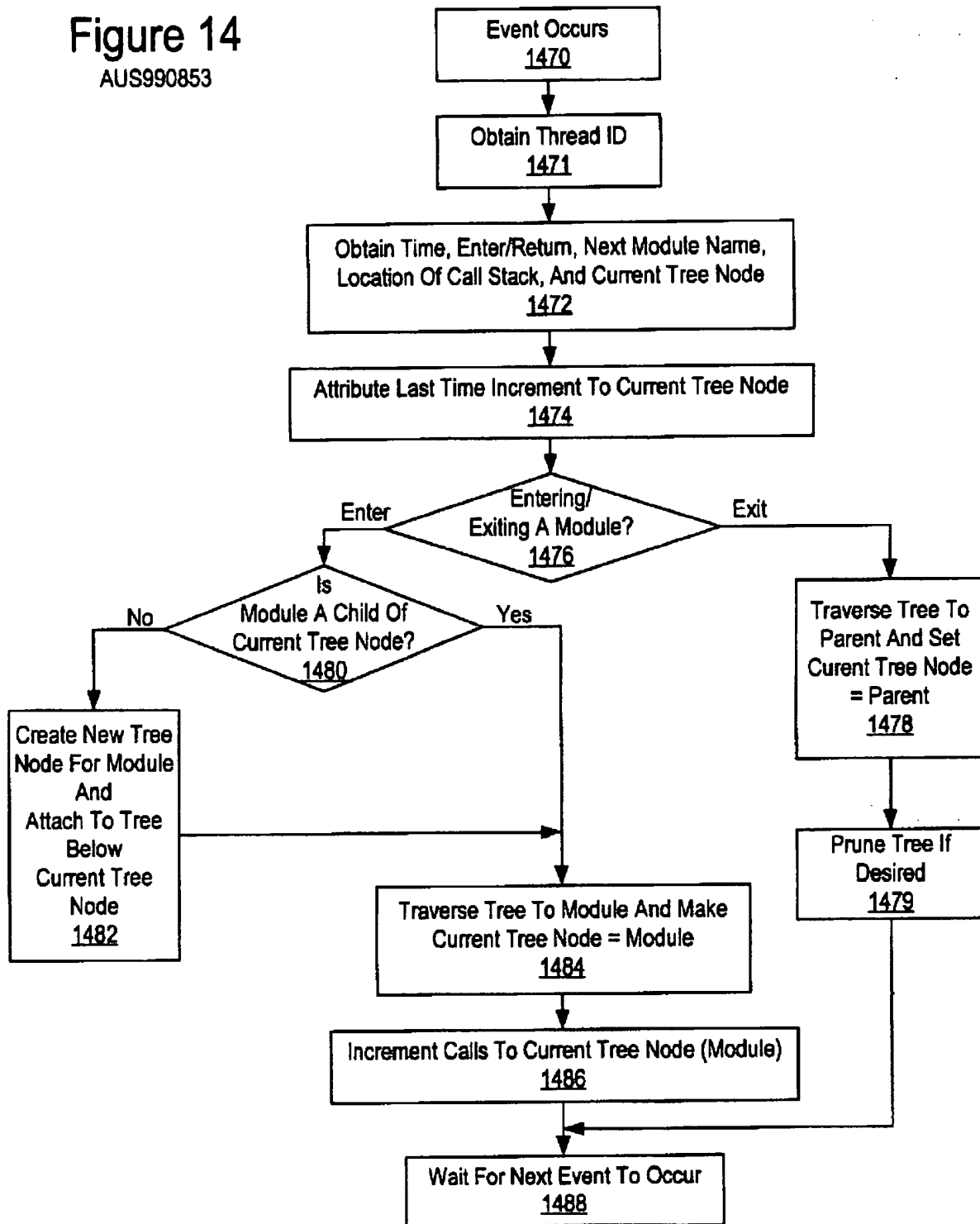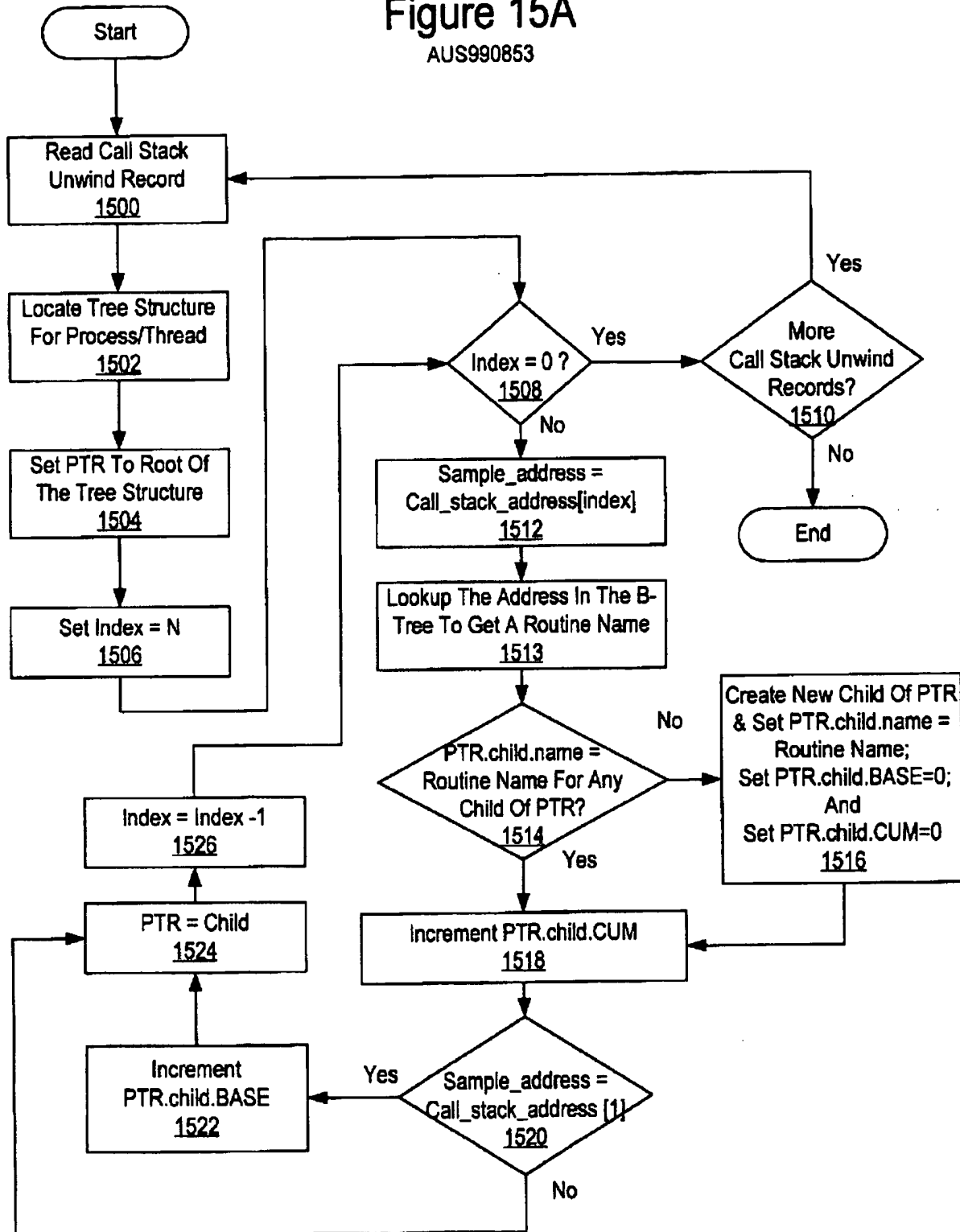| Level 1230 | RL 1232 | Calls 1234 | Base 1236 | Cum 1238 | Indent 1240 |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 19 | pt_pidtid |
| 1 | 1 | 1 | 3 | 19 | - C |
| 2 | 1 | 1 | 3 | 7 | - - A |
| 3 | 1 | 2 | 3 | 4 | - - - B |
| 4 | 2 | 1 | 1 | 1 | - - - - B |
| 2 | 1 | 1 | 2 | 9 | - - B |
| 3 | 1 | 1 | 3 | 7 | - - - A |
| 4 | 2 | 1 | 2 | 3 | - - - - B |
| 5 | 2 | 1 | 1 | 1 | - - - - - A |
| 4 | 1 | 1 | 1 | 1 | - - - - X |

## Figure 13
### AUS990853

```
    ●                                              ●
                                                    │
                                                    │
    ┌───────────────────────────────────────────────┤
    │                                               │
    ▼                                               │
  ╱ More ╲                                          │
 ╱ Trace Records ╲  No ──→ ( Done )                 │
 ╲ To Parse?     ╱                                  │
  ╲  1350  ╱  Yes                                   │
    ╲   ╱    │                                       │
     ▼       ▼                                       │
         ┌──────────────────────────────────┐       │
         │ Obtain Time, Enter/Return, And   │       │
         │ Next Module Name From Trace      │       │
         │ Record                           │       │
         │ 1352                             │       │
         └──────────────────────────────────┘       │
                       │                             │
                       ▼                             │
         ┌──────────────────────────────────┐       │
         │ Attribute Last Time Increment To │       │
         │ Current Tree Node                │       │
         │ 1354                             │       │
         └──────────────────────────────────┘       │
                       │                             │
                       ▼                             │
                  ╱ Entering/ ╲                      │
        Enter    ╱ Exiting A Module? ╲    Exit       │
        ┌───────╲     1356          ╱───────┐        │
        │        ╲                 ╱        │        │
        ▼                                   ▼        │
   ╱ Is ╲                        ┌──────────────────┐│
  ╱ Module A Child Of ╲         │ Traverse Tree To ││
  ╲ Current Tree Node? ╱        │ Parent And Set   ││
   ╲    1360    ╱               │ Current Tree Node││
No ┌──╲      ╱──┐ Yes           │ = Parent         ││
   │   ╲    ╱   │               │ 1358             ├┘
   ▼            ▼               └──────────────────┘
┌────────────┐ ┌──────────────────┐
│Create New  │ │Traverse Tree To  │
│Tree Node   │ │Module And Make   │
│For Module  │ │Current Tree Node │
│And Attach  │ │= Module          │
│To Tree     │ │1364              │
│Below       │ └──────────────────┘
│Current     │        │
│Tree Node   │        ▼
│1362        │ ┌──────────────────┐
└────────────┘ │Increment Calls To│
               │Current Tree Node │
               │(Module)          │
               │1366              │
               └──────────────────┘
```

## Figure 14
AUS990853

Event Occurs
**1470**

↓

Obtain Thread ID
**1471**

↓

Obtain Time, Enter/Return, Next Module Name,
Location Of Call Stack, And Current Tree Node
**1472**

↓

Attribute Last Time Increment To Current Tree Node
**1474**

↓

Entering/
Exiting A Module?
**1476**

Enter ← → Exit

**Enter:**

Is
Module A Child Of
Current Tree Node?
**1480**

No ← → Yes

**No:**

Create New Tree
Node For Module
And
Attach To Tree
Below
Current Tree
Node
**1482**

**Exit:**

Traverse Tree To
Parent And Set
Current Tree Node
= Parent
**1478**

↓

Prune Tree If
Desired
**1479**

**Yes:**

Traverse Tree To Module And Make
Current Tree Node = Module
**1484**

↓

Increment Calls To Current Tree Node (Module)
**1486**

↓

Wait For Next Event To Occur
**1488**

# Figure 15A

AUS990853

```
( Start )
    |
    v
+------------------+
| Read Call Stack  |
| Unwind Record    |<----------------------------------------+
| 1500             |                                          |
+------------------+                                          |
    |                                                         |
    v                                                         |
+------------------+                                          |
| Locate Tree      |                                          |
| Structure        |                                          |
| For Process/     |                          Yes            |
| Thread           |                           ^             |
| 1502             |                           |             |
+------------------+                    +---------------+     |
    |                                   |     More      |     |
    v                                   | Call Stack    |     |
+------------------+        Yes         | Unwind        |-----+
| Set PTR To Root  |    +----------->   | Records?      |
| Of The Tree      |    |               |    1510       |
| Structure        |    |               +---------------+
| 1504             |    |                       |
+------------------+    |                       | No
    |            +-----------+                  v
    v            | Index = 0 ?|             ( End )
+------------------+  1508    |
| Set Index = N    |  No
| 1506             |  |
+------------------+  v
                +---------------------------+
                | Sample_address =          |
                | Call_stack_address[index] |
                | 1512                      |
                +---------------------------+
                            |
                            v
                +---------------------------+
                | Lookup The Address In The |
                | B-Tree To Get A Routine   |
                | Name                      |
                | 1513                      |
                +---------------------------+
                            |
                            v
                +---------------------------+         +---------------------------+
                | PTR.child.name =          |   No    | Create New Child Of PTR   |
                | Routine Name For Any      |-------->| & Set PTR.child.name =    |
                | Child Of PTR?             |         | Routine Name;             |
                | 1514                      |         | Set PTR.child.BASE=0;     |
                +---------------------------+         | And                       |
                            | Yes                     | Set PTR.child.CUM=0        |
                            v                         | 1516                      |
                +---------------------------+         +---------------------------+
                | Increment PTR.child.CUM   |<-------------------+
                | 1518                      |
                +---------------------------+
                            |
                            v
+------------------+  Yes   +---------------------------+
| Increment        |<-------| Sample_address =          |
| PTR.child.BASE   |        | Call_stack_address [1]    |
| 1522             |        | 1520                      |
+------------------+        +---------------------------+
    |                                   | No
    v                                   |
+------------------+                    |
| PTR = Child      |                    |
| 1524             |                    |
+------------------+                    |
    ^                                   |
    |                                   |
+------------------+                    |
| Index = Index -1 |<-------------------+
| 1526             |
+------------------+
```

Begin

Read Value To Obtain
Program Counter Value
1550

End Of File?
1552

Yes

End.

No

Look Up Program Counter
In Global Map
1554

Obtain Process + Function
Id
1556

# Figure 15B

AUS990853

# Figure 16

AUS990853

<u>1600</u>

| Calls<br><u>1602</u> | Base<br><u>1604</u> | Cum<br><u>1606</u> | Name<br><u>1608</u> |
|:---:|:---:|:---:|:---:|
| 1 | 0 | 19 | pt_pit.tid |
| 1 | 3 | 19 | C |
| 3 | 7 | 14 | A |
| 5 | 8 | 13 | B |
| 1 | 1 | 1 | X |

# Figure 17

1700

ArcFlow Output

    Base - Time/Instructions directly in function

    Cum - Time/Instructions directly & indirestly in function

ArcFlow Invarients:

    1) Sum(Parent(Calls)) = Self(Calls)

    2) Sum(Parent(Base)) = Self(Base)

    3) Sum(Parent(Cum)) = Self(Cum)

    4) Sum(Child(Cum)) = Self(Cum) - Self(Base)

| Source | Calls | Base | Cum | Function |
|---|---|---|---|---|
| Self | 1 | 0 | 19 | [0] pt_pidtid |
| Child | 1 | 3 | 19 | C |
| Parent | 1 | 3 | 19 | pt_pidtid |
| Self | 1 | 3 | 19 | [1] C |
| Child | 1 | 2 | 9 | B |
| Child | 1 | 3 | 7 | A |
| Parent | 1 | 3 | 7 | C |
| Parent | 1 | 3 | 7 | B |
| rParent | 1 | 1 | 1 | B |
| Self | 3 | 7 | 17 | [2] A |
|  |  |  | 15 |  |
| Child | 3 | 5 | 7 | B |
| Child | 1 | 1 | 1 | X |
| Parent | 2 | 3 | 4 | A |
| rParent | 1 | 2 | 3 | A |
| Parent | 1 | 2 | 9 | C |
| Self | 5 | 8 | 13 | [3] B |
|  |  |  | 17 |  |
| Child | 1 | 3 | 7 | A |
| rChild | 1 | 1 | 1 | A |
| Child | 1 | 1 | 1 | B |
| Parent | 1 | 1 | 1 | A |
| Self | 1 | 1 | 1 | [4] X |

# Figure 18

AUS990853

Units :: Ticks                                     1800
Total ::    342

| LvL | RL | Calls | Base | Cum | Indent Name |
|-----|-----|-------|------|-----|-------------|
| 1 | 1 | 1 | 0 | 342 | - _Thread-21__(0xe0046618) |
| 2 | 1 | 3 | 0 | 342 | – J:nulltestScore()I |
| 3 | 1 | 2 | 0 | 272 | — J:nulltestMilliseconds(I)I |
| 4 | 1 | 29450 | 0 | 271 | — J:nullexecute()I |
| 5 | 1 | 271 | 0 | 271 | —+ stack_0x40 |
| 6 | 1 | 271 | 0 | 271 | —+- F:ExecuteJava —————————————— 1802 |
| 7 | 1 | 271 | 0 | 271 | —+- F:jit_invokeCompiledEntryMethod |
| 8 | 1 | 271 | 0 | 271 | —+— F:_jit_invokeentry |
| 9 | 1 | 271 | 0 | 271 | —+— F:JITInvokeCompiledEntryMethod_md |
| 10 | 1 | 271 | 0 | 271 | —+—+ J:nullrun()V |
| 11 | 2 | 271 | 0 | 271 | —+—+- J:nulltestScore()I |
| 12 | 2 | 271 | 0 | 271 | —+—+— J:nulltestMilliseconds(I)I |
| 13 | 2 | 271 | 268 | 271 | —+—+— J:nullexecute()I |
| 14 | 1 | 2 | 0 | 2 | —+—+— F:jperf_methodEntry |
| 15 | 1 | 2 | 0 | 2 | —+—+—+ F:SoftTracehook |
| 16 | 1 | 2 | 2 | 2 | —+—+—+- F:enable_Interrupts |
| 14 | 1 | 1 | 1 | 1 | —+—+— F:jperf_methodExit |
| 4 | 1 | 1 | 0 | 1 | — stack_0x40 |
| 5 | 1 | 1 | 0 | 1 | —+ F:ExecuteJava —————————————— 1804 |
| 6 | 1 | 1 | 0 | 1 | —+- F:jit_invokeCompiledEntryMethod |
| 7 | 1 | 1 | 0 | 1 | —+- F:_jit_invokeentry |
| 8 | 1 | 1 | 0 | 1 | —+— F:JITInvokeCompiledEntryMethod_md |
| 9 | 1 | 1 | 0 | 1 | —+— J:nullrun()V |
| 10 | 2 | 1 | 0 | 1 | —+—+ J:nulltestScore()I |
| 11 | 2 | 1 | 0 | 1 | —+—+- J:nulltestMilliseconds(I)I |
| 12 | 1 | 1 | 0 | 1 | —+—+- J:nullexecute()I |
| 13 | 1 | 1 | 0 | 1 | —+—+— F:jperf_methodExit |
| 14 | 1 | 1 | 0 | 1 | —+—+— F:SoftTracehook |
| 15 | 1 | 1 | 1 | 1 | —+—+—+ F:enable_interrupts |
| 4 | 1 | 2 | 0 | 0 | — J:nullcleanUp()I |

1806

1808

# Figure 19

AUS990853

| Major Code | Minor Code | Data Item 1 | Data Item 2 | Data Item 3 | Data Item 4 | Data Item 5 | Description |
|---|---|---|---|---|---|---|---|
| 0x4 | level + 0x1 | depth | n/a | n/a | n/a | | begin interrupt at level |
| 0x4 | level + 0x80000000 1 | depth | n/a | n/a | n/a | | end interrupt at level |
| 0x10 | 0xab | system tid | java tid | is System Thread (boolean) | n/a | | thread created without a name while trace active |
| 0x10 | 0xac | system tid | n/a | n/a | n/a | | identifies the idle thread |
| 0x10 | 0xad | system tid | n/a | n/a | n/a | | identifies the garbage collection thread |
| 0x10 | 0xae | system tid | java tid | thread name | n/a | | thread created with a name while trace active |
| 0x30 | 0x10 | object id | method block address | n/a | n/a | | method invocation (interpreted) |
| 0x30 | 0x10 + 0x80000000 | object id | method block address | n/a | n/a | | method exit (interpreted) |
| 0x40 | 0x7ffffff | number (n) of stack unwinds at timer interrupt | pc1-program counter of interrupted routine | pc2-caller of interrupted routine | . . . . . . . | pcn-1 of n-2nd caller of interrupted routine | pcn of n-1st caller of interrupted routine |
| 0x41 | 0x7ffffff | number (n) of stack unwinds at instrumented routine | pc1-program counter of instrumented routine | pc2-caller of instrumented routine | . . . . . . . | | pcn of n-1st caller of instrumented routine |
| 0x50 | 0x10 | object id | method block address | n/a | n/a | | method invocation (jitted) |
| 0x50 | 0x10 + 0x80000000 0 | object id | method block address | n/a | n/a | | method exit (jitted) |

# Figure 20

## Application Level Thread Oriented Approach

# Figure 21A
## Application Level
### Thread Oriented Approach

```
    ( Begin )
        |
        v
  +-----------+
  | JVM Called|
  |   2102    |
  +-----------+
        |
        v
+-------------------------+
| JVM Initializes Profiler Via The JVM |
| Profiler Interface By Sending A List |
|      Of Metrics To Track             |
|            2104                      |
+-------------------------+
        |
        v
+-------------------------+
| The Profiler Allocates Data Areas For: |
|          Total (M);                    |
|        Cum (M,t); And                  |
|        LastCum (M,t),                  |
|      On Current Thread (t)             |
|            2106                        |
+-------------------------+
        |
        v
     ( End )
```
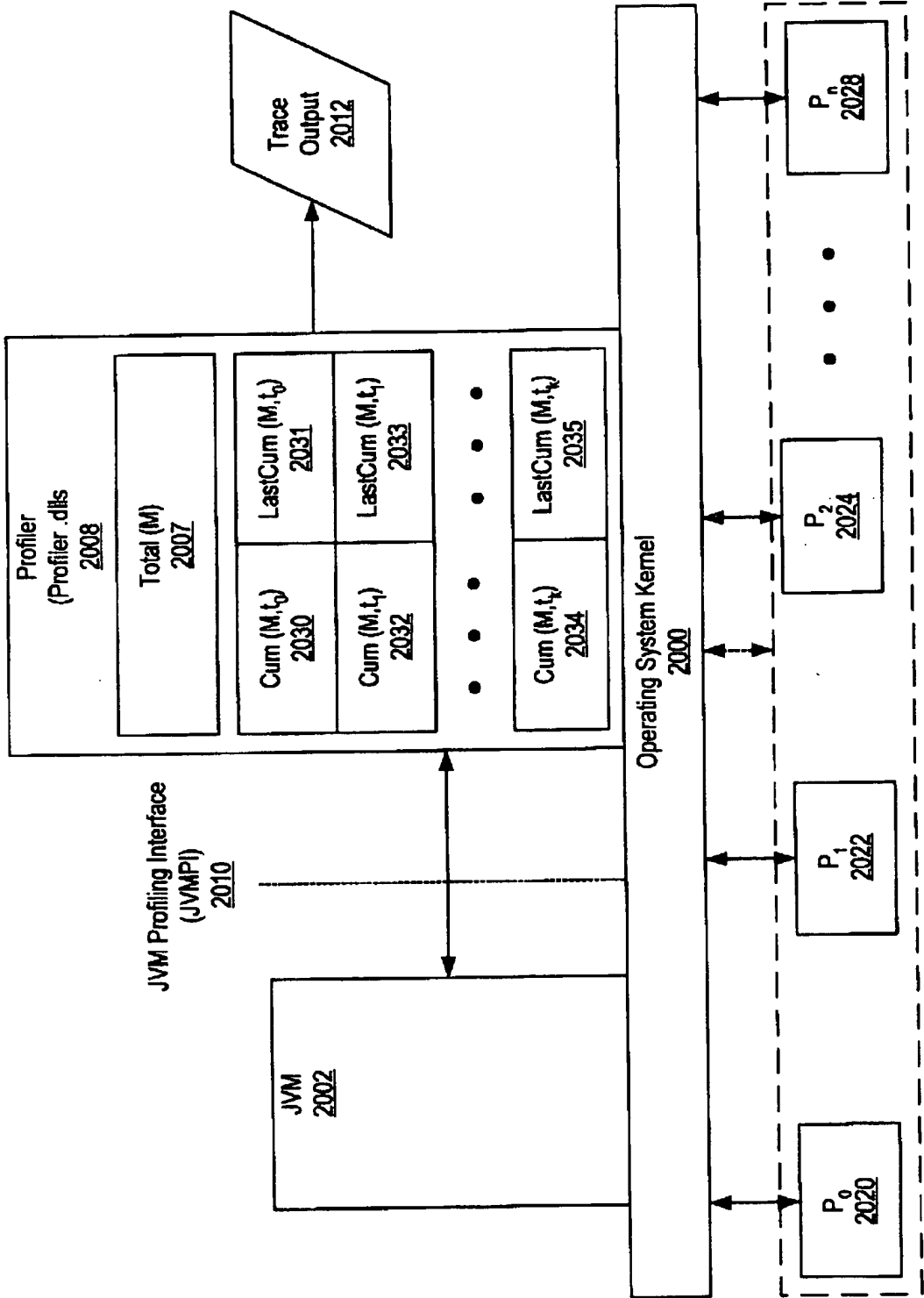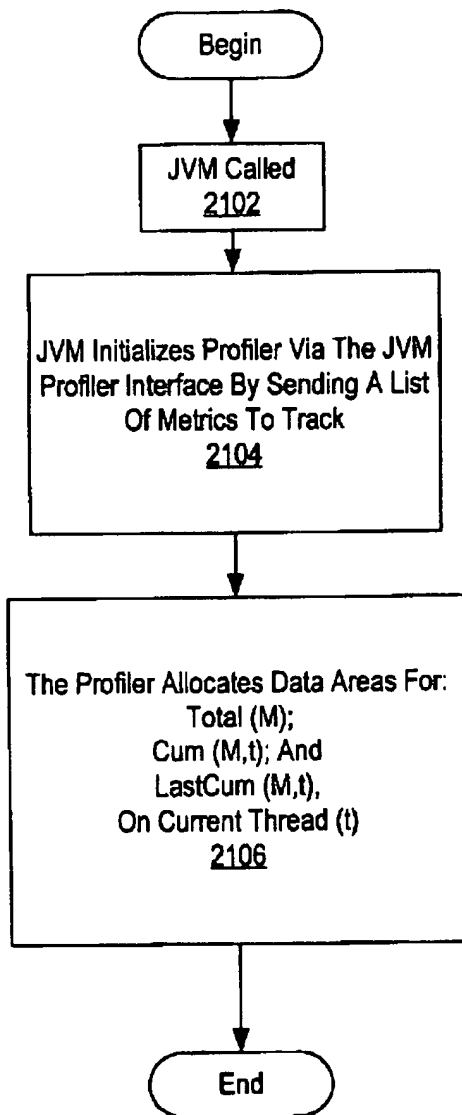
# Figure 21B
## Application Level
### Thread Oriented Approach for Each New Thread

```
    ( Begin )
        |
        v
+-------------------------+
| The Profiler Allocates Data Areas For: |
|      Cum (M,$t_{new}$); And            |
|      LastCum (M,$t_{new}$),            |
|      On New Thread ($t_{new}$)         |
|            2110                        |
+-------------------------+
        |
        v
     ( End )
```
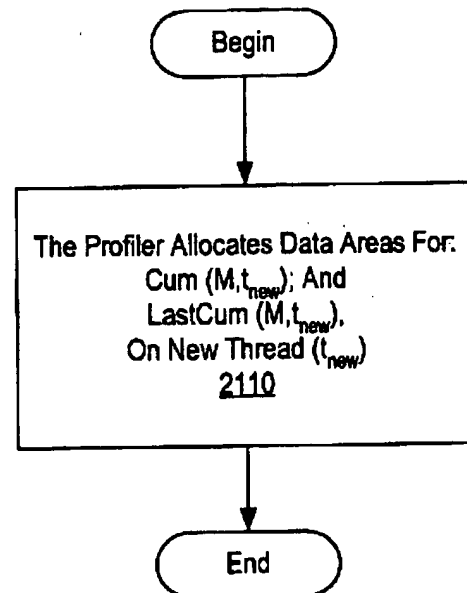
# Figure 22

Profiler Receives Metric Event from Jvm

```
        ┌─────────────┐
        │    Begin    │
        └─────────────┘
               │
               ▼
   ┌───────────────────────┐
   │ Jvm Has A Metric Event │
   │         2202          │
   └───────────────────────┘
               │
               ▼
   ┌───────────────────────┐
   │ Jvm Sends Event To Profiler │
   │         2204          │
   └───────────────────────┘
               │
               ▼
   ┌───────────────────────┐
   │ Profiler Receives Metric Event, │
   │        MDelta (m)     │
   │         2206          │
   └───────────────────────┘
               │
               ▼
┌─────────────────────────────────┐
│ Profiler Updates Appropriate Total And Cum │
│ Values i.e., Total (m) And Cum (m,t), │
│      Total (m) += MDelta (m)    │
│      Cum (m,t) += MDelta (m)    │
│              2208               │
└─────────────────────────────────┘
               │
               ▼
        ┌─────────────┐
        │     End     │
        └─────────────┘
```

# Figure 23

Application Level Thread Oriented Approach

```
              ┌──────────┐
              │  Begin   │
              └────┬─────┘
                   │
                   ▼
┌──────────────────────────────────────────────────────┐
│                                                        │
│  Profiler Receives Via The Jvm Profiler Interface An   │
│         Entry Or Exit Event On Running Thread (t)      │
│                        2302                            │
│                                                        │
└──────────────────────┬─────────────────────────────────┘
                       │
                       ▼
┌──────────────────────────────────────────────────────┐
│                                                        │
│            Profiler Calculates Delta Value:            │
│      Delta (M) = Cum (M,t) - LastCum (M,t)             │
│                        2304                            │
│                                                        │
└──────────────────────┬─────────────────────────────────┘
                       │
                       ▼
┌──────────────────────────────────────────────────────┐
│                                                        │
│            Profiler Updates LastCum To Cum:            │
│            LastCum (M,t) = Cum (M,t)                    │
│                        2306                            │
│                                                        │
└──────────────────────┬─────────────────────────────────┘
                       │
                       ▼
┌──────────────────────────────────────────────────────┐
│            Method (M,t) += Delta (M)                   │
│                        2308                            │
└──────────────────────┬─────────────────────────────────┘
                       │
                       ▼
                  ┌──────────┐
                  │   End    │
                  └──────────┘
```

# Figure 24

API for Metric Values

```
            ( Begin )
                |
                v
+-------------------------------------------+
| Jvm Requests Cum (M,t) OR Total (M) From  |
|               Profiler                    |
|                 2402                       |
+-------------------------------------------+
                |
                v
+-------------------------------------------+
|     Profiler Returns Requested Value(s)    |
|                 2404                        |
+-------------------------------------------+
                |
                v
             ( End )
```

## Figure 25
AUS990853

JVM Profiling Interface (JVMPI) 2510

JVM 2502

Operating System Kernel 2500

**per Processor Cum Metric Values**

| per Processor Cum Metric Values | per Processor Last Cum Metric Values |
|---|---|
| PCum (M,$p_0$) 2530 | PLastCum (M,$p_0$) 2540 |
| PCum (M,$p_1$) 2532 | PLastCum (M,$p_1$) 2542 |
| PCum (M,$p_2$) 2534 | PLastCum (M,$p_2$) 2544 |
| • • • | • • • |
| PCum (M,$p_n$) 2538 | PLastCum (M,$p_n$) 2548 |

$P_0$ 2520
$P_1$ 2522
$P_2$ 2524
$P_n$ 2528

**per Thread Cum Metric Values**

| per Thread Cum Metric Values |
|---|
| TCum (M,$t_0$) 2560 |
| TCum (M,$t_1$) 2562 |
| • • • |
| TCum (M,$t_k$) 2568 |

Profiler (Profiler .dlls) 2508

**Profiler Cum Metric Values**

| Profiler Cum Metric Values |
|---|
| ProfCum (M,$t_0$) 2570 |
| ProfCum (M,$t_1$) 2572 |
| • • • |
| ProfCum (M,$t_k$) 2574 |

Trace Output 2512

# Figure 26A

AUS990853

Initialization Process

Begin

Jvm Called
2602

Jvm Passes Parameters To The Profiler
2604

Profiler Calls Jvm To Indicate Metrics Being Tracked
2606

Profiler Allocates And Initializes:
ProfCum (M,t) = 0
2608

Kernel Allocates And Initializes:
PCum (M,p) = 0, For Each Processor;
PLastCum(M,p)=0, For Each Processor;
And
TCum (M,t) = 0, For Current Thread (t)
2610

End
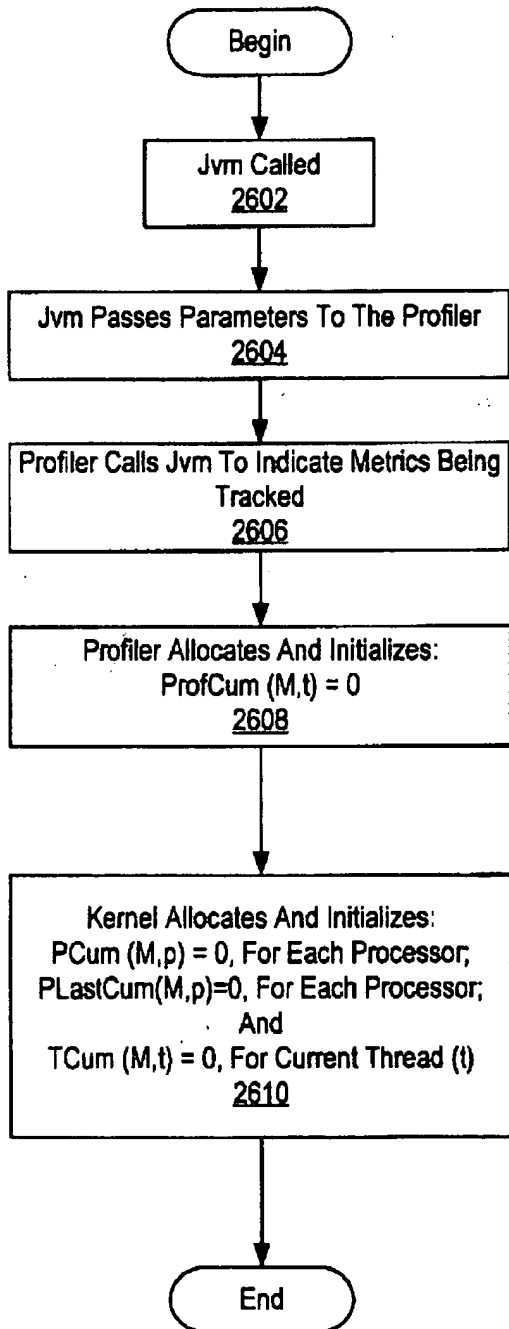
# Figure 26B

AUS990853

Process for the Kernel Recognizing a New Thread on Processor (p)

Begin

Kernel Allocates And Initializes:
PCum (M,p) = 0;
PLastCum(M,p)=0; And
TCum (M,t) = 0, For Current Thread (t)
2620

End
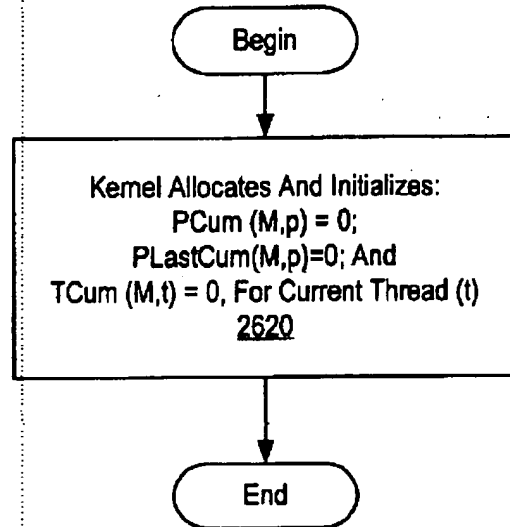
# Figure 26C

AUS990853

Process for the Profiler Recognizing a New Thread

Begin

Profiler Allocates And Initializes:
ProfCum (M,t) = 0
2630

End
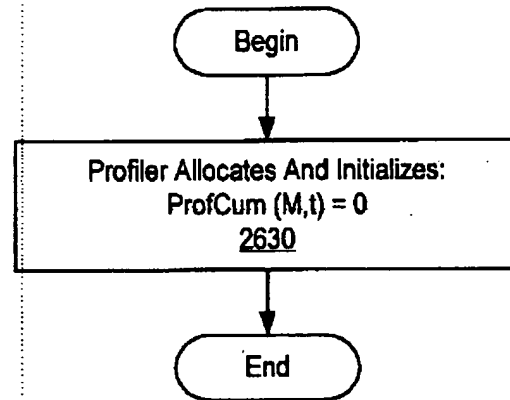
# Figure 27

## Jvm Process for Updating Variable Values
## for Metrics Being Tracked

```
              ( Begin )
                  │
                  ▼
┌─────────────────────────────────────────┐
│ The Value Of A Metric That Is Being      │
│ Tracked Changes                          │
│ MDelta (m)                               │
│ 2702                                     │
└─────────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────────┐
│ Jvm Updates Total (m):                   │
│ Total (m) += MDelta (m)                  │
│ 2704                                     │
└─────────────────────────────────────────┘
                  │
                  ▼
┌─────────────────────────────────────────┐
│ Jvm Updates PCum (m,p):                  │
│ PCum (m,p) += MDelta (m)                 │
│ 2706                                     │
└─────────────────────────────────────────┘
                  │
                  ▼
               ( End )
```
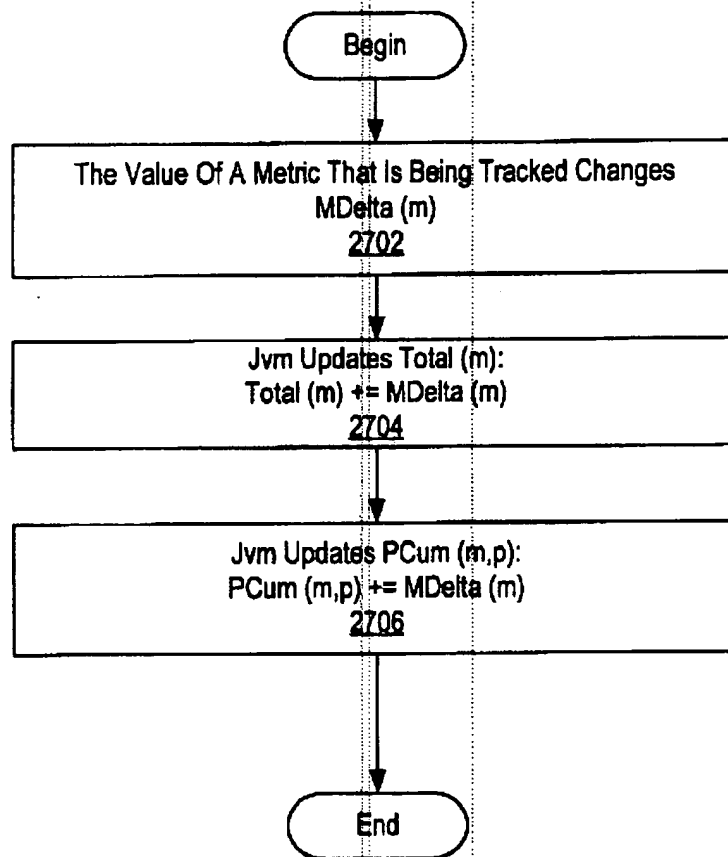
## Figure 28

### Process for the Jvm Directly Updating Per Processor Global Variable Values for Metrics Being Tracked

Begin

A Metric That Is Being Tracked
Changes, MDelta (m)
2802

The Jvm's Register Allows Direct Update Of The Per Processor
Data Area In A Single Atomic Instruction (AOOI):
PCum (m,p) += MDelta (m)
2804

End
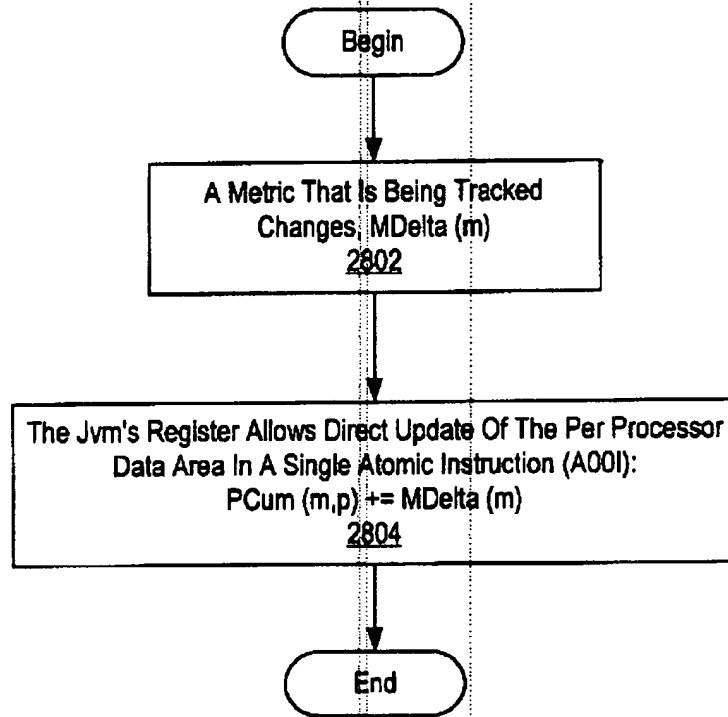
# Figure 29

## Process for the OS Kernel Updating Per Processor Global Variable Values for Metrics Being Tracked

Begin

A Metric That Is Being Tracked For A Thread
Changes, MDelta (m)
2902

The Jvm Requests The Kernel Change The Metric
2904

Kernel Ensures That The Metric Value Is Updated
Without Releasing Current Processor:
PCum (m,p) += MDelta (m)
2906

End
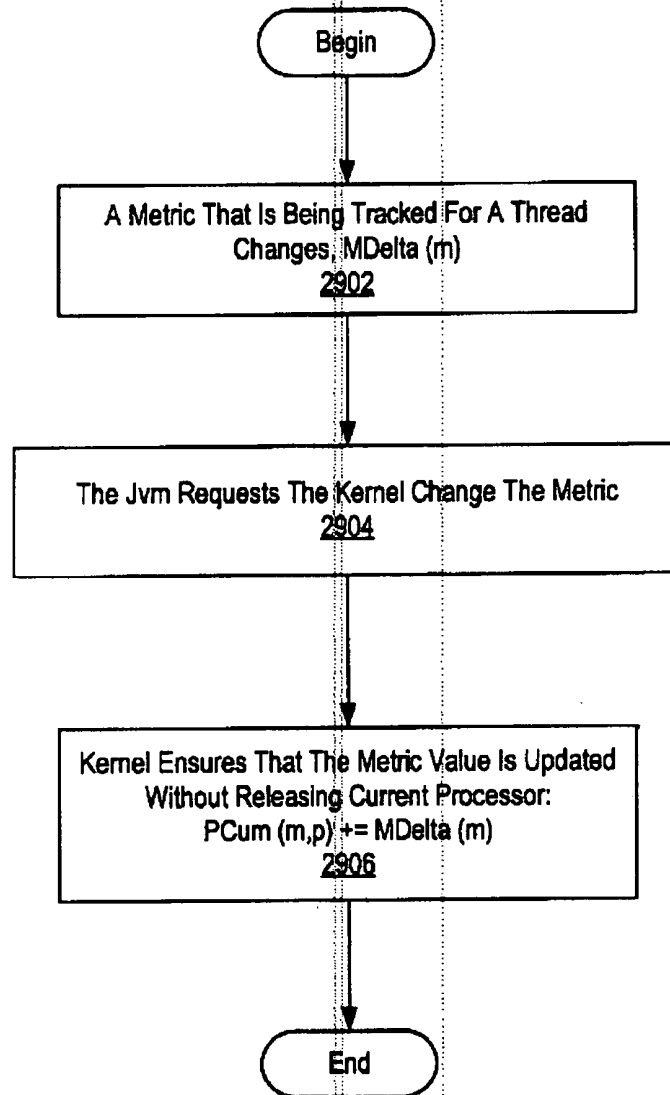
# Figure 30

## Process for the Kernel Updating Base Metric Variable Values for Metrics Being Tracked in Response to a Thread Dispatch Event

```
                    ┌─────────────┐
                    │    Begin    │
                    └─────────────┘
                           │
                           ▼
    ┌──────────────────────────────────────────────┐
    │  Thread ($t_{old}$) On Processor (p) Is        │
    │  Stopped And Thread ($t_{new}$) Is Dispatched  │
    │  To Processor (p)                              │
    │                   3002                         │
    └──────────────────────────────────────────────┘
                           │
                           ▼
    ┌──────────────────────────────────────────────┐
    │  The Kernel Calculates The Change, Delta (M):  │
    │  Delta (M)= PCum (M,p) - PLastCum (M,p)        │
    │                   3004                         │
    └──────────────────────────────────────────────┘
                           │
                           ▼
    ┌──────────────────────────────────────────────┐
    │  Kernel Updates PLastCum:                      │
    │  PLastCum (M,p) = PCum (M,p)                    │
    │                   3006                         │
    └──────────────────────────────────────────────┘
                           │
                           ▼
    ┌──────────────────────────────────────────────┐
    │  Kernel Updates TCum:                          │
    │  TCum ($M,t_{old}$) += Delta (M),              │
    │  For Thread ($t_{old}$)                         │
    │                   3008                         │
    └──────────────────────────────────────────────┘
                           │
                           ▼
                    ┌─────────────┐
                    │     End     │
                    └─────────────┘
```
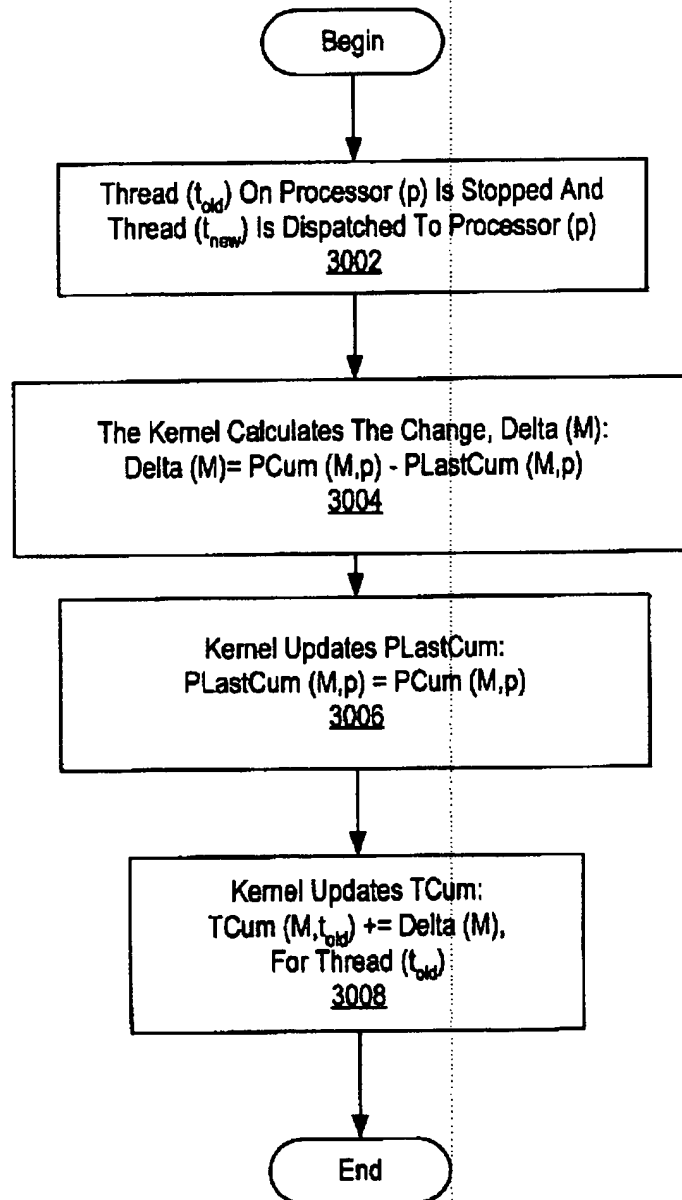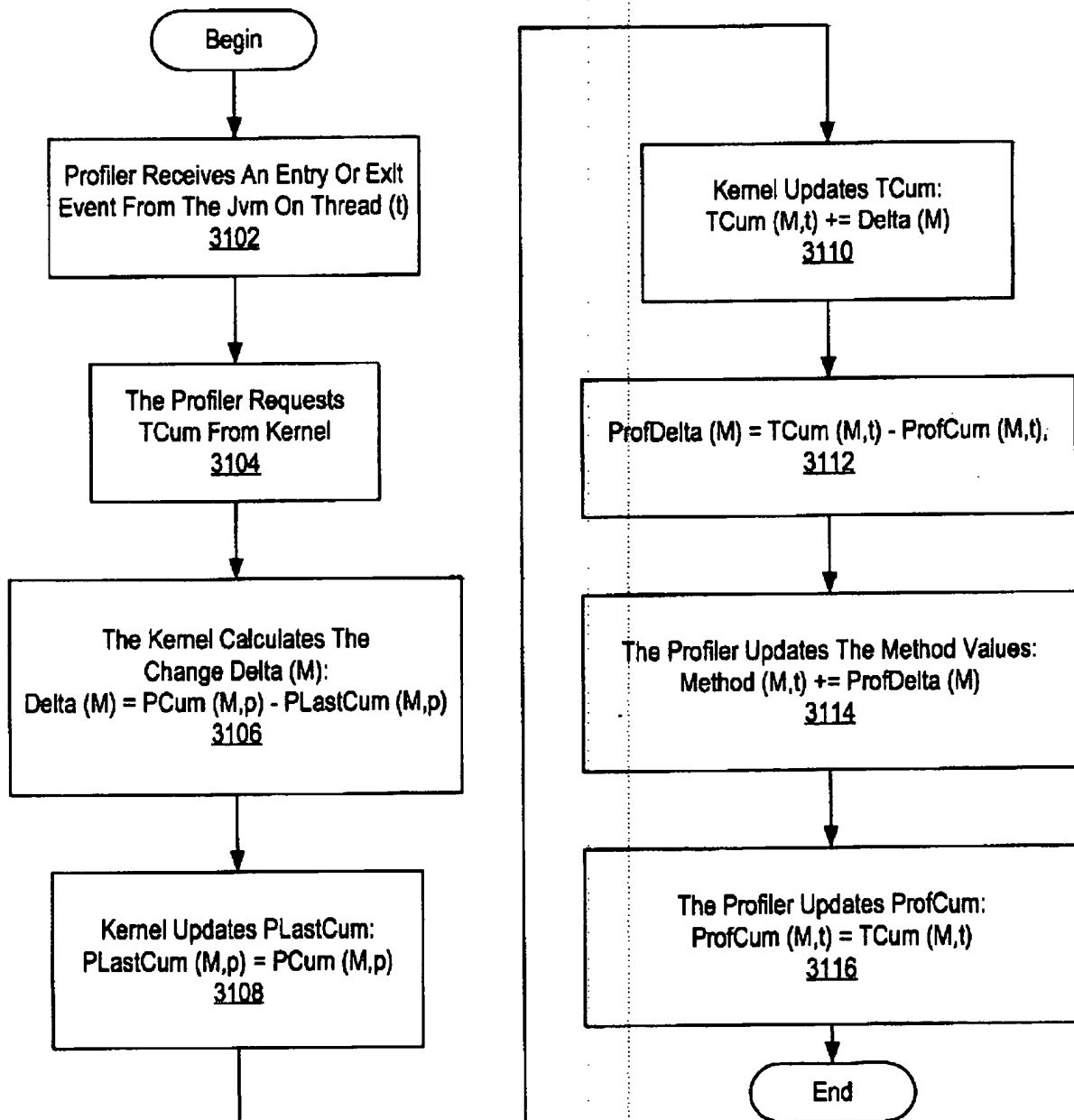
# Figure 31

AUS990853

Process for the Profiler Updating Base Metric Variable Values for
Metrics Being Tracked in Response to a Method Entry or Exit Event

```
                    ( Begin )

   ┌─────────────────────────┐          ┌─────────────────────────┐
   │ Profiler Receives An     │          │ Kernel Updates TCum:     │
   │ Entry Or Exit Event      │          │ TCum (M,t) += Delta (M)  │
   │ From The Jvm On Thread(t)│          │ 3110                     │
   │ 3102                     │          └─────────────────────────┘
   └─────────────────────────┘                     │
              │                                     │
   ┌─────────────────────────┐          ┌─────────────────────────┐
   │ The Profiler Requests    │          │ ProfDelta (M) =          │
   │ TCum From Kernel         │          │ TCum (M,t) - ProfCum(M,t)│
   │ 3104                     │          │ 3112                     │
   └─────────────────────────┘          └─────────────────────────┘
              │                                     │
   ┌─────────────────────────┐          ┌─────────────────────────┐
   │ The Kernel Calculates The│          │ The Profiler Updates The │
   │ Change Delta (M):        │          │ Method Values:           │
   │ Delta (M) = PCum (M,p) - │          │ Method (M,t) += ProfDelta(M)│
   │ PLastCum (M,p)           │          │ 3114                     │
   │ 3106                     │          └─────────────────────────┘
   └─────────────────────────┘                     │
              │                          ┌─────────────────────────┐
   ┌─────────────────────────┐          │ The Profiler Updates     │
   │ Kernel Updates PLastCum: │          │ ProfCum:                 │
   │ PLastCum (M,p) = PCum(M,p)│         │ ProfCum (M,t) = TCum(M,t)│
   │ 3108                     │          │ 3116                     │
   └─────────────────────────┘          └─────────────────────────┘
              │                                     │
              └──────────────────────┘          ( End )
```